SQLAlchemy-Defaults Documentation

Release 0.2

Konsta Vesterinen

June 03, 2016

Contents

1 What does it do? 3

SQLAlchemy-Defaults is a plugin for SQLAlchemy that provides smart defaults for lazy guys, like me.

CHAPTER 1

What does it do?

- By setting default values for your int/str/bool columns, SQLAlchemy-Defaults automatically also sets server_default values
- All string columns are not nullable by default.
- Unlike SQLAlchemy, all boolean columns are not nullable and False by default.
- Provides auto_now feature for datetime columns
- Automatically assigns names for enum types which doesn't have the name set
- Easy min/max check constraints based on min and max column info arguments
- Auto creates indexes for foreign key columns (very useful when using PostgreSQL)

So instead of writing this:

```
from datetime import datetime
import sqlalchemy as sa


class User(Base):
    id = sa.Column(sa.Integer, primary_key=True)

    name = sa.Column(
        sa.Unicode(255),
        nullable=False
    )
    description = sa.Column(
        sa.Unicode(255),
        nullable=False,
        default=u'',
        server_default=u''
    )

    is_admin = sa.Column(
        sa.Boolean,
        default=False,
        server_default=sa.sql.expression.false(),
        nullable=False
    )

    created_at = sa.Column(
        sa.DateTime,
        default=datetime.utcnow,
```

```
        server_default=sa.func.now(),
    )

    hobbies = sa.Column(
        sa.Integer,
    )

    __table_args__ = (
        sa.schema.CheckConstraint(
            'user.hobbies >= 1'
        ),
        sa.schema.CheckConstraint(
            'user.hobbies <= 4'
        )
    )
```

You can simply write (notice here how we define an empty __lazy_options__ dict):

```
import sqlalchemy as sa
from sqlalchemy_defaults import Column


class User(Base):
    __lazy_options__ = {}
    id = Column(sa.Integer, primary_key=True)

    name = Column(
        sa.Unicode(255),
    )
    description = Column(
        sa.Unicode(255),
        default=u'',
    )

    is_admin = Column(
        sa.Boolean,
    )

    created_at = Column(
        sa.DateTime,
        auto_now=True
    )

    hobbies = Column(
        sa.Integer,
        min=1,
        max=4
    )
```

After you've defined all your models you need to make desired mapper lazy configured (or you can simple make all mappers lazy configured by passing sa.orm.mapper as the first argument):

```
from sqlalchemy_defaults import make_lazy_configured


make_lazy_configured(sa.orm.mapper)
```